# OPTIMAL DESIGN OF TRUSSED STRUCTURES.

by

William Arthur Crockett

United States
Naval Postgraduate School

# THESIS

OPTIMAL DESIGN OF TRUSSED STRUCTURES

by

William Arthur Crockett



October 1969

# Optimal Design of Trussed Structures

by

William Arthur Crockett
Major, United States Army
B.S., University of Texas, 1960

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
October 1969

## ABSTRACT

Structural design problems can be considered to be optimization problems because a design is sought which is optimal by some criterion subject to limitations on size, behavior, or other aspects of the structure. Under certain conditions, such problems may be solved by conventional mathematical programming techniques. The minimum weight design of a statically indeterminate three-bar truss is used to illustrate the application of the "sequential unconstrained minimization technique" of Fiacco and McCormick to the optimal design of trussed structures. A suggested computational procedure and computer program are included.

TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF ILLUSTRATIONS

## I. INTRODUCTION

The desired conclusion of the structural design process is a structural configuration which performs its intended function efficiently. The method by which the final design is determined usually includes the establishment of an initial configuration from which the final design is evolved through a process of analysis and redesign.

The electronic computer has permitted an acceleration of the design process by enabling the rapid analysis of the mathematical models associated with the structures under consideration. In order to more efficiently use this capability L. Schmit [1] has suggested a rational approach to the design process called "systematic structural synthesis." This approach involves the systematic evaluation and modification of a large number of trial designs until the optimal design is obtained.

The basic problem is to design a structure which is optimal by some criterion subject to a set of requirements which specify acceptable limits on the behavior, size, or other aspects of the structure. Such problems are known to the operations analyst as mathematical programming problems, the general form of which is as follows. Find a vector X which

$$\begin{aligned} &\text{minimizes } F(X), \\ &\text{subject to } H_i(X) = 0, \; i = 1,2,\ldots,m; \\ &\qquad\qquad\quad G_j(X) \geq 0, \; j = 1,2,\ldots,p. \end{aligned}$$

Here, $X = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional column vector; $F(X)$ is the objective function; and the relations $H_i(X) = 0$ and $G_j(X) \geq 0$ represent the constraints on the problem.

If $F(X)$ is convex in $X$ and the constraint region is convex, i.e. the set of all solutions to $H_i(X) = 0$, $i = 1, \ldots, m$ and $G_j(X) = 0$, $j = 1, \ldots, p$ forms a convex set, the problem is called a convex programming problem. If a problem can be classified as a convex programming problem then there are solution procedures which are guaranteed to find the global optimum.

Several techniques which have been devised to solve this special case of the general programming problem are Rosen's "gradient projection method," Zoutendijk's "method of feasible directions," and Kelly's "cutting-plane" method. These procedures are summarized by Hadley in Ref. 7. Another convex programming technique called the "method of alternate steps" was suggested by Schmit [1] as a means for converging on the optimal structural design.

A recently developed convex programming technique which appears to be especially promising is the "sequential unconstrained minimization technique for convex programming with equality constraints" of Fiacco and McCormick [2]. By this method, the objective function and constraints are dealt with simultaneously by combining them into a single function. This function is minimized for different values of an auxiliary parameter thereby generating a sequence of feasible solutions that converge to the optimal solution.

If the convexity restrictions for the constraint region or the objective function are relaxed a convex programming solution technique may still be used but there is no guarantee that a solution to the problem is other than a local minimum. For most practical problems, it is no less valuable to obtain information about local optima in the absence of a global solution.

In general, structural design problems do not meet the convexity restrictions for convex programming problems. One such problem is the determination of the cross-sectional areas of the members of a statically indeterminant, coplanar, three-bar truss such that the total weight of the truss is minimized subject to constraints on stress and displacement. This structure is illustrated in Figure 1. Procedures which can be used to solve this problem can also be applied to other trussed structures.

## II. OBJECTIVE AND SCOPE

The purpose of this study was to illustrate the application of the Fiacco-McCormick technique to the optimal design of trussed structures. Since minimum weight has usually been the major goal in the development of aircraft structural design techniques, it was used as the criterion for optimality.

One example of the determination of the minimum weight design of a three-bar truss is included with results and the computer program. Also included is a brief discussion and outline of the solution procedure used to solve the problem. This procedure can only be applied to problems for which constraints can be formulated in completely analytical form.

## III. THE THREE-BAR TRUSS PROBLEM

The three-bar truss problem of Schmit [1] is illustrated in the following sketch.

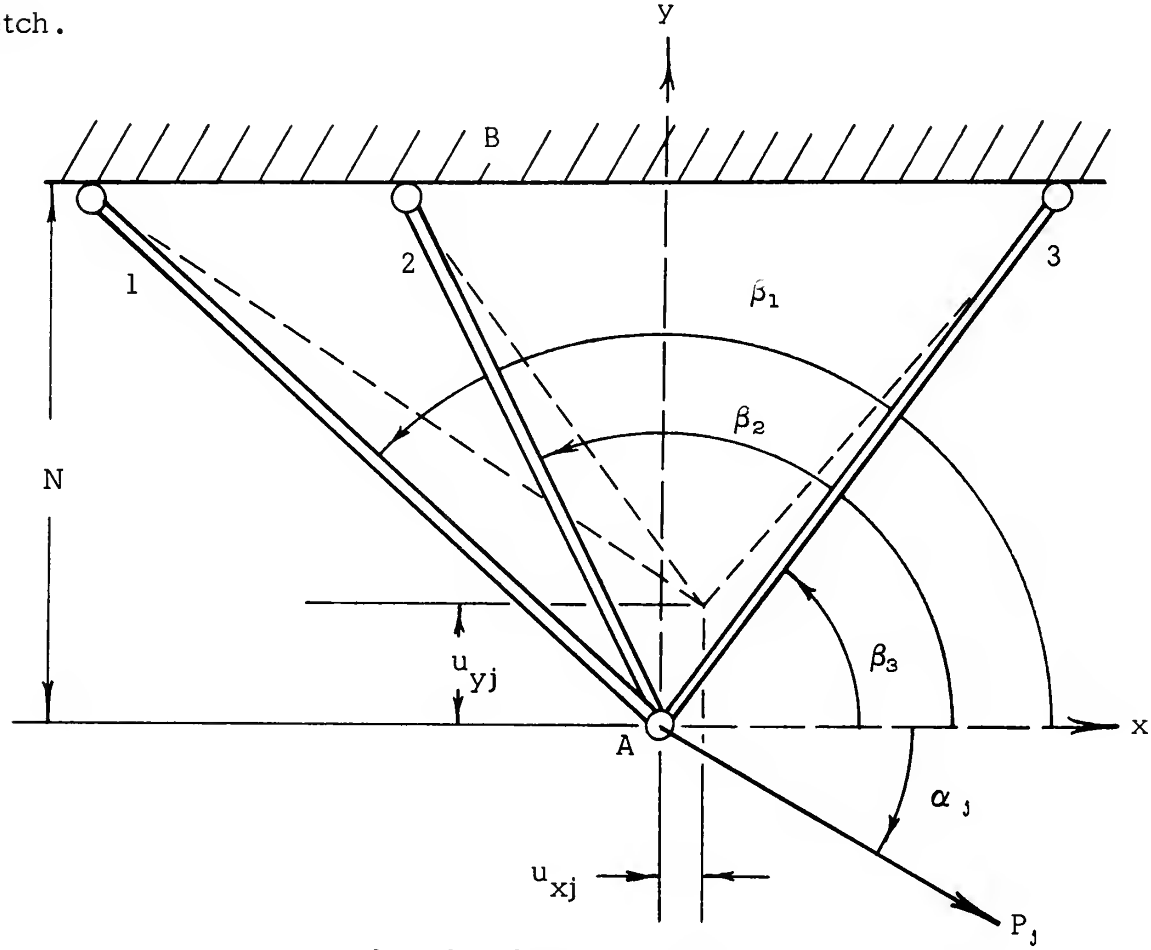


Figure 1. The Three-Bar Truss

The function of the truss is to transmit the load $P_j$ from the point of application at joint A to the support B. The deflections of the joint A in the x and y directions due to the load $P_j$ are $u_{xj}$ and $u_{yj}$ respectively.

For a coplanar system of forces, one of the conditions for equilibrium of a rigid body in one plane is that the algebraic sum of the components of all forces in each of two mutually perpendicular directions in the plane of the forces must separately vanish. Therefore,

$$\Sigma F_x = p_{1j} \cos\beta_1 + p_{2j} \cos\beta_2 + p_{3j} \cos\beta_3 + P_j \cos\alpha_j = 0$$

and

$$\Sigma F_y = p_{1j} \sin\beta_1 + p_{2j} \sin\beta_2 + p_{3j} \sin\beta_3 - P_j \sin\alpha_j = 0$$

where $p_{ij}$ is the force component in the direction of the ith structural member due to the jth load $P_j$ .

The stress in the ith member due to the jth force is defined as

$$\sigma_{ij} = p_{ij} / A_i$$

where $A_i$ is the cross-sectional area of the ith member. The equilibrium equations can then be written as

$$A_1\sigma_{1j} \cos\beta_1 + A_2\sigma_{2j} \cos\beta_2 + A_3 \sigma_{3j} \cos\beta_3 + P_j \cos\alpha_j = 0 \qquad (1)$$

and

$$A_1 \sigma_{1j} \sin\beta_1 + A_2\sigma_{2j} \sin\beta_2 + A_3 \sigma_{3j} \sin\beta_3 - P_j \sin\alpha_j = 0 \qquad (2)$$

Assuming small displacements, the change in length of the ith structural member due to the jth load is

$$\delta_{ij} = - u_{xj} \cos\beta_i - u_{yj} \sin\beta_i \ . \qquad (3)$$

A relationship between stess, displacement, and temperature change can be established if is noted that $\delta_{ij}$ may also be expressed as

$$\delta_{ij} = \frac{p_{ij}\ l_i}{A_i\ E_i} + \overline{\alpha}_i \Delta T_i\ l_i \ . \qquad (4)$$

where $l_i$ is the length of the ith member, $\overline{\alpha}_i$ is the mean coefficient of thermal expansion of the ith member, $E_i$ is the modulus of elasticity of the ith member, and $\Delta T_i$ is the temperature change imposed externally on the ith member.

If the temperature is assumed to be constant, equations (3) and (4) can be combined to yield

$$\frac{p_{ij} l_i}{A_i E_i} + u_{xj} \cos\beta_i + u_{uj} \sin\beta_i = 0$$

and since $\sigma_{ij} = p_{ij}/A_i$ and $l_i = N/\sin\beta_i$ , this equation becomes

$$\frac{\sigma_{ij} N}{E_i \sin\beta_i} + u_{xj} \cos\beta_i + u_{yj} \sin\beta_i = 0. \tag{5}$$

For purposes of this illustration, it is assumed that all geometric parameters of the truss are given except for the cross-sectional areas. It is further assumed that in addition to non-negativity restrictions for the $A_i$ values, the behavioral limits for the structure are specified by upper and lower bounds on the stresses and displacements. Then, for the jth load condition,

$$L_{ij} \le \sigma_{ij} \le U_{ij} \;, \; i = 1,2,3,$$

$$L_{xj} \le u_{xj} \le U_{xj},$$

and

$$L_{yj} \le u_{yj} \le U_{yj}.$$

If $\rho_i$ denotes the specified density of the material in the ith structural member then the weight, W, of the truss is

$$W = A_1 \, l_1 \, \rho_1 + A_2 \, l_2 \, \rho_2 + A_3 \, l_3 \, \rho_3$$

Since $l_i = N/\sin\beta_i$ , this equation can be rewritten as

$$W = \frac{A_1 \; N \rho_1}{\sin\beta_1} + \frac{A_2 \; N \; \rho_2}{\sin\beta_2} + \frac{A_3 \; N \; \rho_3}{\sin\beta_3}$$

which is linear in A and is therefore convex.

The mathematical programming problem is then to fine $A_1$ , $A_2$ , $A_3$ , the cross-sectional areas of the structural members which minimize

$$W = A_1 \frac{N \cdot \rho_1}{\sin\beta_1} + A_2 \frac{N \cdot \rho_2}{\sin\beta_2} + A_3 \frac{N \cdot \rho_3}{\sin\beta_3}$$

subject to the constraints

$$A_1\sigma_{1j} \cos\beta_1 + A_2\sigma_{2j} \cos\beta_2 + A_3\sigma_{3j} \cos\beta_3 + P_j \cos\alpha_j = 0$$

$$A_1\sigma_{1j} \sin\beta_1 + A_2\sigma_{2j} \sin\beta_2 + A_3\sigma_{3j} \sin\beta_3 - P_j \sin\alpha_j = 0$$

$$\sigma_{1j} \frac{N}{E_1 \sin\beta_1} + u_{xj}\cos\beta_1 + u_{xj}\sin\beta_1 = 0$$

$$\sigma_{2j} \frac{N}{E_2 \sin\beta_2} + u_{xj}\cos\beta_2 + u_{yj}\sin\beta_2 = 0$$

$$\sigma_{3j} \frac{N}{E_3 \sin\beta_3} + u_{xj}\cos\beta_3 + u_{yj}\sin\beta_3 = 0$$

and

$$L_{ij} \le \sigma_{ij} \le U_{ij}, \; L_{xj} \le u_{xj} \le U_{xj}, \; L_{yj} \le u_{yj} \le U_{yj}, \; A_i \ge 0$$

for $i = 1,2,3$ and $j = 1,2,\ldots,k$ where k denotes the number of distinct loading conditions. It is assumed that the loading conditions are not applied simultaneously. Note that inequalities such as $L_{ij} \le \sigma_{ij} \le U_{ij}$ can be rewritten as

$$\sigma_{ij} - L_{ij} \ge 0,$$

$$-\sigma_{ij} + U_{ij} \ge 0.$$

Now, in this problem statement, the variables to be determined are actually $A_1, A_2, A_3, \sigma_{ij}$, $i=1,2,3$, $j=1,\ldots,k$ and $u_{xj}$, $u_{yj}$, $j = 1,\ldots,k$. It is therefore apparent that the first two equality constraints are non-linear because of the cross products $\sigma_{ij} A_i$ and hence the constraint region is non-convex.

## IV. SOLUTION PROCEDURE

Recall that the general form of the mathematical programming problem is to determine the vector X which

$$\text{minimizes} \quad F(X),$$

$$\text{subject to} \quad H_i(X) = 0, \; i = 1,2,\ldots,m;$$

$$G_j(X) \geq 0, \; j = 1,2,\ldots,p,$$

where $X^T = (x_1, x_2, \ldots, x_n)^T$ is an n-dimensional column vector.

The basis of the Fiacco and McCormick technique is the P-function which is defined as follows for this general form.

$$P(X,r) = F(X) + r \sum_{i=1}^{m} \frac{1}{G_i(X)} + r^{-\frac{1}{2}} \sum_{j=1}^{p} (H_j(X))^2 .$$

The general solution procedure is then to seek minimum values of P(X,r) for $r = r_1, r_2, \ldots, r_k$ where $r_1 > r_2 \ldots r_k > 0$. Then

$$\lim_{k \to \infty} X_k = \bar{X}$$

and

$$\lim_{k \to \infty} P(X_k, r_k) = \bar{F}$$

where $\bar{F}$ is the minimum value of the objective function of the mathematical programming problem at $\bar{X}$.

The conditions for the determination of a global minimum for the P function are generally the same as for the convex programming problem. The objective function, F(X), must be a convex function. Recall that a linear function is convex although not strictly convex. Also the constraint region must be convex (which is not true for this problem) and $F, G_1, \ldots, G_m, H_1, \ldots, H_p$ must be continuous.

Close inspection of the P function reveals that the term

$$r \sum_{i=1}^{m} \frac{1}{G_i(X)}$$

is a boundary repulsion term which keeps the minimum feasible solution of the P-function in the interior of the region defined by the inequality constraints. Naturally, the use of such a boundary repulsion term requires that solutions exist in the interior of the region defined by the inequality constraints.

It is this feature of the Fiacco-McCormick technique which provides a significant improvement over other convex programming techniques. Other approaches, including Schmit's "method of alternate steps" require elaborate techniques to determine what action to take when the boundary is encountered during the minimization process.

Another consequence of the boundary repulsion term which should be noted, however, is that the search for a feasible P-function minimum must be confined to the region defined by the inequality contraints. The reason for this restriction is apparent from the following example. Suppose that $G_1(X) = x_1 \geq 0$. As $x_1 \to 0$ from the negative side $[G_1(X)]^{-1} \to -\infty$.

The P-function can be minimized by direct analytical procedures for relatively trivial cases only. For other cases, some method of successive approximations must be used. One such method is that of gradient descent.

A. GRADIENT DESCENT

Gradient methods are based on a Taylor's series approximation of the function to be minimized.

1. First-Order Gradient Method

The first-order gradient method uses only the first-order terms of the Taylor's series expansion to obtain the equation,

$$X_2 = X_1 - \theta \nabla P(X_1)$$

The procedure is then to move a distance proportional to $\theta$ in the direction of the gradient vector evaluated at the point $X_1$. $X_2$ is the point at which the P-function is minimized along the gradient.

2. Second-Order Gradient Method

A better approximation of the P-function is obtained by using the second-order terms of the Taylor's series expansion as well as the first to obtain the equation

$$X_2 = X_1 - \theta \left\| \frac{\partial^2 P(X_1)}{\partial x_i \partial x_j} \right\|^{-1} \nabla P(X_1)$$

where the matrix $\|\cdot\|$ is the Hessian matrix evaluated at $X_1$ and -1 denotes the matrix inverse. The P-function is minimized along the vector given by the product of the inverse of the Hessian and the gradient of the function evaluated at $X_1$.

If the function to be minimized is not strictly convex, the Hessian matrix may be positive definite only in the vicinity of a local minimum. For this reason, the procedure used began with a search in the direction of positive $\theta$. If the P-function did not decrease in this direction before encountering the boundary, the search was continued in the direction of negative $\theta$. This situation actually occurred in the example when the P-function was minimized at r=1.

3. Higher-Order Methods

It would seem logical to consider the use of higher order terms to obtain an even better approximation of the function to be minimized. The effort required to program such a procedure, however, would be prohibitive for all but trivial problems. In fact, the labor required to calculate the second-order partial derivatives for the Hessian matrix may prohibit the use of the second-order gradient method for large problems. Experience with several small examples using both techniques indicates that the convergence properties of the second-order technique are far superior to those of the first-order technique. Fiacco and McCormick provide examples of both techniques in Ref. 3 which clearly demonstrate the superiority of the second-order method.

## B. ESTIMATION OF THE FINAL P-FUNCTION MINIMUM

Fiacco and McCormick show in Ref. 2 that estimates of the solution of the P-function minimum for r=0 can be obtained by the use of polynomial approximations of $x_1, x_2, \ldots, x_n$ and P as a function of $r_k^{1/2}$ as follows. Expand each component of the X vector associated with the minimum of P(X,r) for a given value of r in a power series in $r^{1/2}$. To obtain a (p-1) th-order polynomial approximation, drop the terms of the power series after the pth term. If the P-function has been minimized for $r_k$, k=1,...,h where $h \geq p$, form the linear equations:

$$x_1(r_{h-p}) = a_o + \ldots + a_{j-1}(r_{h-p})^{\frac{j-1}{2}} + \ldots + a_{p-1}(r_{h-p})^{\frac{p-1}{2}}$$

$$\vdots \qquad\qquad\qquad\qquad \vdots$$

$$x_1(r_h) = a_o + \ldots + a_{j-1}(r_h)^{\frac{j-1}{2}} + \ldots + a_{p-1}(r_h)^{\frac{p-1}{2}}$$

If it is assumed that r is reduced by a constant factor such that $r_{k+1} = r_k / C$ or $r_k = r_1 / C^{k-1}$, then the above equations constitute a system of p linear equations in p unknowns, $a_o, \ldots, a_{p-1}$. For the polynomial,

$$x_1 = a_o + a_1 r + a_2 r^2 + \ldots + a_{p-1} r^{p-1} ,$$

$x_1$ at r=0 is given by $a_o$. Hence, if these equations are solved for $a_o$, a (p-1) th-order polynomial approximation is obtained for the value of $x_1$ corresponding to the solution to the mathematical programming problem. Estimates are obtained by this procedure for $x_1, \ldots, x_n$ as well as P at r=0. Because of computer accuracy limitations, the order of the polynomial estimate will be restricted to three and will use the last four data points.

## C. CRITERIA FOR TERMINATION

As r approaches zero, the value of the P-function minimum approaches the optimal solution. More rapid convergence to the optimal solutiom may be obtained by the extrapolation described above thereby permitting termination of the procedure with fewer iterations. The specific point at which the minimization process is to terminate depends, of course, upon the accuracy desired of the final solution estimate.

Fiacco and McCormick suggest several alternate criteria in Ref. 2 based on the changes in the values of the variables or functions from one P-minimum to the next. In the case of the estimated value of the P-function minimum, the minimization process is terminated when

$$|P^*(r_{i-1}) - P^*(r_i)| < \epsilon$$

where $\epsilon$ is a small positive number and $P^*(r_i)$ is the polynomial estimate of the final solution after the P-function is minimized for the ith r value. For the three-bar truss problem, $\epsilon$ was chosen to be $1 \times 10^{-5}$.

## D. SUMMARY OF SOLUTION PROCEDURE

The following procedures are those required to obtain an estimated value for the optimal solution to the mathematical programming problem.

1. Form the P-function and obtain the analytical forms for the gradient vector and Hessian matrix.

2. Set p=4 for a third-order polynomial estimate of the values of P and $x_1, \ldots, x_n$ for r=0.

3. Set r=1 and h=1 and select a starting point, $X_0$, such that the inequality constraints are satisfied.

4. Determine the direction of steepest descent at the starting point. Search along this direction in the region defined by the inequality constraints until the directional derivative of the P-function is less than the specified tolerance, i.e. near zero.

5. Test the gradient magnitude. If less than a specified tolerance, go to step 8.

6. Call the present point the new starting point and go to step 4.

7. If $h \geq p$ calculate the $(p-1)$ th-order polynomial estimate of the components of the solution at $r=0$.

8. If $h \geq p+1$ compare the last two estimates of the P function at $r=0$. If within the specified tolerance, terminate the routine. Otherwise, set $r = r/C$ where C is the reduction factor and set $h = h+1$. Go to step 6.

# V. THE EXAMPLE

The example used to illustrate the application of the Fiacco-McCormick technique to the design of trussed structures is the problem described previously with two distinct load conditions. This yields a problem with 13 variables; 10 equality constraints, 4 of which are non-linear; and 23 inequality constraints including the three non-negativity constraints for the cross-sectional areas of the truss members.

## A. PROGRAMMING CONSIDERATIONS

The computer program was written in Fortran IV language for the IBM 360 computer at the Naval Postgraduate School in Monterey, California. No attempt was made to optimize the program to reduce execution time. As it is written, the program required approximately 25 seconds for execution.

A simple bracketing procedure was used to locate the minimum of the P-function along the vector in the direction of steepest descent. The directional derivative was evaluated at the current starting point. A $\epsilon$ was then determined corresponding to a point at which the directional derivative was opposite in sign from that at the starting point and still within the region defined by the inequality constraints. The bracketing procedure was then commenced to reduce the magnitude of the directional derivative below the specified tolerance, DDTOL.

Some experimentation was required to determine the necessary magnitude of DDTOL to ensure convergence of the gradient magnitude to within a tolerance of EPSI = $1 \times 10^{-3}$. This was determined to be DDTOL = $1 \times 10^{-7}$.

For the three-bar truss problem, a double-precision version of the program was required although several two-variable problems were solved without recourse to double precision. For small r values, the P-function becomes progressively more peaked in the vicinity of the minimum value. For the three-bar truss problem this peakedness is such that the change in $\theta$ required to obtain a bracket small enough to yield a directional derivative within tolerance is below the accuracy of the computer unless double precision is used.

For programming purposes, the notation of the problem variables $W, A_1, A_2, A_3, \sigma_{11}, \sigma_{21}, \sigma_{31}, u_{x1}, u_{y1}, \sigma_{12}, \sigma_{22}, \sigma_{32}, u_{x2}, u_{y2}$ was changed to $F, x_1, x_2, \ldots, x_{13}$ respectively.

B. INPUT DATA

The input data for the problem was that of Schmit's Case 1 in Ref. 1. A cursory inspection will reveal that the values are highly unrealistic since the actual range of values of the modulus of elasticity is from 5 to $30 \times 10^6$ lbs./in.$^3$ for metals. They are, however, sufficient for purposes of illustrating the solution technique. The input data is as follows.

| | | | |
|---|---|---|---|
| $P_1 = 30$ | $\rho_1 = 1$ | $\beta_1 = 135°$ | $E_1 = 1$ |
| $P_2 = 20$ | $\rho_2 = 1$ | $\beta_2 = 90°$ | $E_2 = 1$ |
| $\alpha_1 = 60°$ | $\rho_3 = 1$ | $\beta_3 = 45°$ | $E_3 = 1$ |
| $\alpha_2 = 180°$ | | | |
| $N = 1$ | | | |

The constraints specified for the problem are

$$A_1, A_2, A_3 \geq 0 ,$$

$$-15 \leq \sigma_{1j}, \sigma_{2j}, \sigma_{3j} \leq 20, \quad j=1,2,$$

and

$$-150 \leq u_{xj}, u_{yj} \leq 200, \quad j=1,2.$$

The starting point for the example was arbitrarily chosen to be $X^T = (2,2,2,5,5,5,5,5,5,5,5,5,5)^T$. The reduction factor for r was 10 at each iteration, i.e. $r_{i+1} = r_i/10$.

C. RESULTS

Values obtained for minimum P as a function of r as well as the third-order estimates of the P-function minimum for r=0 are shown in Table I. The value of the objective function, F, corresponding to the indicated r value; the total value of the remaining terms of the P function, denoted by P-F; and the values of the equality constraints, $H_i(X)$ are also shown.

The value given under "Moves Required to Minimize" is the number of times a new direction of steepest descent was calculated prior to achieving a gradient magnitude of less than .001 for the current r value.

The worth of the polynomial estimate is indicated by the fact that a final solution estimate of minimum P is obtained on the sixth iteration which differs by less than $1\times10^{-4}$ from the value obtained on the ninth and final iteration.

By the ninth iteration, the value of the objective function, F, has converged to within $7\times10^{-5}$ of the final solution estimate while the value of $P(r_9)$ differs by $14\times10^{-5}$ from the final estimate.

Table I. Computer Solution for P-Function, Objective Function and Constraints

| LINE | R | MOVES REQUIRED TO MINIMIZE | P | F | P-F | H1 | H2 | H3 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.000000000 | 37 | 8.21205 | 4.40535 | 3.80670 | 0.00914 | -0.00711 | 0.00030 |
| 2 | 0.100000000 | 12 | 3.88535 | 3.18278 | 0.70257 | 0.00171 | -0.00586 | 0.00138 |
| 3 | 0.010000000 | 6 | 3.12510 | 2.99796 | 0.12714 | 0.00018 | -0.00229 | 0.00066 |
| 4 | 0.001000000 | 4 | 2.97467 | 2.94523 | 0.02943 | -0.00006 | -0.00075 | 0.00025 |
| | 3RD ORDER ESTIMATE | | 2.92127 | | | | | |
| 5 | 0.000100000 | 4 | 2.93733 | 2.92936 | 0.00797 | -0.00005 | -0.00024 | 0.00009 |
| | 3RD ORDER ESTIMATE | | 2.92179 | | | | | |
| 6 | 0.000010000 | 2 | 2.92690 | 2.92457 | 0.00233 | -0.00002 | -0.00008 | 0.00003 |
| | 3RD ORDER ESTIMATE | | 2.92231 | | | | | |
| 7 | 0.000001000 | 2 | 2.92379 | 2.92308 | 0.00071 | -0.00001 | -0.00002 | 0.00001 |
| | 3RD ORDER ESTIMATE | | 2.92238 | | | | | |
| 8 | 0.000000100 | 2 | 2.92283 | 2.92260 | 0.00022 | -0.00000 | -0.00001 | 0.00000 |
| | 3RD ORDER ESTIMATE | | 2.92239 | | | | | |
| 9 | 0.000000010 | 2 | 2.92253 | 2.92246 | 0.00007 | -0.00000 | -0.00000 | 0.00000 |
| | 3RD ORDER ESTIMATE | | 2.92239 | | | | | |

Table I. (Continued)

| LINE | R | H4 | H5 | H6 | H7 | H8 | H9 | H10 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.000000000 | -0.000424 | 0.000299 | -0.027515 | 0.005707 | 0.004153 | -0.005879 | 0.004162 |
| 2 | 0.100000000 | -0.001948 | 0.001377 | -0.013061 | -0.001148 | -0.000459 | 0.000649 | -0.000459 |
| 3 | 0.010000000 | -0.000932 | 0.000659 | -0.004277 | -0.000717 | -0.000279 | 0.000395 | -0.000279 |
| 4 | 0.001000000 | -0.000349 | 0.000247 | -0.001397 | -0.000186 | -0.000072 | 0.000102 | -0.000072 |
| 5 | 0.000100000 | -0.000127 | 0.000089 | -0.000459 | -0.000040 | -0.000015 | 0.000022 | -0.000015 |
| 6 | 0.000010000 | -0.000042 | 0.000030 | -0.000147 | -0.000011 | -0.000004 | 0.000006 | -0.000004 |
| 7 | 0.000001000 | -0.000013 | 0.000009 | -0.000047 | -0.000003 | -0.000001 | 0.000002 | -0.000001 |
| 8 | 0.000000100 | -0.000004 | 0.000003 | -0.000015 | -0.000001 | -0.000000 | 0.000001 | -0.000000 |
| 9 | 0.000000010 | -0.000001 | 0.000001 | -0.000005 | -0.000000 | -0.000000 | 0.000000 | -0.000000 |

Table II. Computer Solution for Cross-Sectional Areas, Stress, and Displacement.

| LINE | R | X1 | X2 | X3 | X4 | X5 | X6 | X7 |
|---|---|---|---|---|---|---|---|---|
| 1 | 1.000000000 | 1.34826 | 1.09069 | 0.99556 | 14.34603 | 12.47881 | -1.86637 | 16.21240 |
| 2 | 0.100000000 | 1.17904 | 0.59013 | 0.65425 | 18.09910 | 18.29197 | 0.19676 | 17.90234 |
| 3 | 0.010000000 | 1.10696 | 0.55190 | 0.62267 | 19.29345 | 19.52305 | 0.23146 | 19.06199 |
| 4 | 0.001000000 | 1.08143 | 0.54629 | 0.61488 | 19.70304 | 19.85547 | 0.15314 | 19.54990 |
| | 3RD ORDER EST | 1.06913 | 0.54445 | 0.61135 | 19.89930 | 20.00018 | 0.10094 | 19.79835 |
| 5 | 0.000100000 | 1.07397 | 0.54460 | 0.61231 | 19.82623 | 19.95593 | 0.12996 | 19.69627 |
| | 3RD ORDER EST | 1.07067 | 0.54380 | 0.61109 | 19.88166 | 20.00169 | 0.12006 | 19.76161 |
| 6 | 0.000010000 | 1.07189 | 0.54402 | 0.61142 | 19.86164 | 19.98624 | 0.12468 | 19.73696 |
| | 3RD ORDER EST | 1.07098 | 0.54374 | 0.61098 | 19.87729 | 19.99997 | 0.12267 | 19.75462 |
| 7 | 0.000001000 | 1.07126 | 0.54383 | 0.61112 | 19.87244 | 19.99565 | 0.12324 | 19.74919 |
| | 3RD ORDER EST | 1.07097 | 0.54374 | 0.61098 | 19.87736 | 19.99998 | 0.12262 | 19.75474 |
| 8 | 0.000000100 | 1.07106 | 0.54377 | 0.61103 | 19.87583 | 19.99863 | 0.12280 | 19.75303 |
| | 3RD ORDER EST | 1.07097 | 0.54374 | 0.61099 | 19.87740 | 20.00000 | 0.12260 | 19.75481 |
| 9 | 0.000000010 | 1.07100 | 0.54375 | 0.61100 | 19.87691 | 19.99957 | 0.12266 | 19.75425 |
| | 3RD ORDER EST | 1.07097 | 0.54374 | 0.61099 | 19.87740 | 20.00000 | 0.12260 | 19.75481 |

Table II. (Continue)

| LINE | R | X8 | X9 | X10 | X11 | X12 | X13 |
|---|---|---|---|---|---|---|---|
| 1 | 1.000000000 | -12.47923 | -11.37516 | 1.57935 | 12.96627 | -24.34142 | -1.58523 |
| 2 | 0.100000000 | -18.29391 | -13.69592 | 4.82715 | 18.52177 | -32.21769 | -4.82651 |
| 3 | 0.010000000 | -19.52398 | -14.54279 | 5.01896 | 19.56096 | -34.10376 | -5.01857 |
| 4 | 0.001000000 | -19.85582 | -14.86148 | 4.99726 | 19.85854 | -34.72002 | -4.99716 |
| | 3RD ORDER EST | -20.00021 | -15.01923 | 4.97391 | 19.99328 | -35.01251 | -4.97398 |
| 5 | 0.000100000 | -19.95606 | -14.95890 | 4.99531 | 19.95417 | -34.91307 | -4.99529 |
| | 3RD ORDER EST | -20.00171 | -15.00297 | 4.99590 | 19.99889 | -35.00185 | -4.99591 |
| 6 | 0.000010000 | -19.98628 | -14.98729 | 4.99810 | 19.98537 | -34.97266 | -4.99809 |
| | 3RD ORDER EST | -19.99997 | -14.99991 | 5.00010 | 20.00001 | -34.99992 | -5.00010 |
| 7 | 0.000001000 | -19.99567 | -14.99599 | 4.99938 | 19.99537 | -34.99136 | -4.99938 |
| | 3RD ORDER EST | -19.99998 | -14.99997 | 5.00004 | 20.00001 | -34.99998 | -5.00004 |
| 8 | 0.000000100 | -19.99863 | -14.99873 | 4.99980 | 19.99853 | -34.99727 | -4.99980 |
| | 3RD ORDER EST | -20.00000 | -15.00000 | 5.00000 | 20.00000 | -35.00000 | -5.00000 |
| 9 | 0.000000010 | -19.99957 | -14.99960 | 4.99994 | 19.99954 | -34.99914 | -4.99994 |
| | 3RD ORDER EST | -20.00000 | -15.00000 | 5.00000 | 20.00000 | -35.00000 | -5.00000 |

One consequence of not obtaining an absolutely null gradient vector is revealed by the values of the equality constraints. Since convergence is obtained only to a gradient magnitude of .001, the equality constraints are not exactly satisfied for large r values. The values of the equality constraints are seen to decrease, however, at each iteration. In other words, the constraints get progressively "tighter" as $r \to 0$. The reason for this is the peakedness of the P-function in the vicinity of the minimum as r gets smaller and smaller. As P becomes more peaked, it is necessary to get nearer the minimum to satisfy the gradient magnitude tolerance for convergence. Thus the approximations become more accurate as $r \to 0$. A gradient magnitude tolerance of $1 \times 10^{-4}$ failed to yield a change in the value of the constraints of sufficient magnitude to change the results although it did require a slightly higher execution time.

Table II shows the values of the successive $x_i(r)$, $i=1,\ldots,13$ as well as the final solution estimates. The final solution estimates changed by less than $2 \times 10^{-4}$ from the sixth to the ninth iteration.

The values obtained by the method of Fiacco and McCormick are compared with those obtained by Schmit [1] using his "method of alternate steps" in Table III.

TABLE III

| Variable | Fiacco & McCormick | Alternate Steps |
|---|---|---|
| $A_1$ | 1.071 | 1.072 |
| $A_2$ | .544 | .544 |
| $A_3$ | 611 | .611 |
| $\sigma_{11}$ | 19.877 | 19.863 |
| $\sigma_{21}$ | 20.000 | 19.984 |
| $\sigma_{31}$ | .123 | 1.206 |
| $u_{x1}$ | 19.755 | 19.743 |
| $u_{y1}$ | 20.000 | -19.984 |
| $\sigma_{12}$ | -15.000 | -14.993 |
| $\sigma_{22}$ | 5.000 | 5.003 |
| $\sigma_{32}$ | 20.000 | 19.996 |
| $u_{x2}$ | -35.000 | -34.998 |
| $u_{y2}$ | - 5.000 | - 5.003 |
| W | 2.922 | 2.924 |

The results obtained by the two methods are quite close although the minimum weight determined by the method of Fiacco and McCormick was slightly less. The primary reason for the difference was probably the termination criterion. Schmit solved the problem in 1960 on an IBM 653 digital computer with severe storage capacity limitations so the accuracy of the solution in this study is unquestionably higher. Advances in hardware make any comparison of execution times meaningless. Schmit's results for $\sigma_{31}$ is erroneous and is probably a typographical error in the reference.

# VI. DISCUSSION

## A. ALTERNATE MINIMA

A solution to the three-bar truss problem has been determined but the fact remains that it can be verified to be only a local minimum. Procedures exist by which the "goodness" of the solution may be determined but the effort may be more than that required for solving the entire problem by a trial and error solution technique. One such procedure is simply to vary the starting point and search for alternate minima.

The example problem was solved for several other starting points selected at random. These included

$$\begin{aligned} X_0 &= (1.0,1.0,1.0,5.0,5.0,\ldots,5.0)^T \\ &= (0.5,0.5,0.5,0.0,0.0,\ldots,0.0)^T \\ &= (0.25,0.25,0.25,0.0,\ldots,0.0)^T \\ &= (0.25,0.25,0.25,-5.0,\ldots,-5.0)^T \\ &= (0.25,0.25,0.25,-10.0,\ldots,-10.0)^T \end{aligned}$$

The only change in value noted was a difference in the number of moves required to minimize the P-function for r=1.

## B. TWO-BAR TRUSS

If the center bar of the truss is removed, the structure becomes a statically determinant two-bar truss in which the internal forces are uniquely determined. The minimum weight of such a structure subject to the constraints of the example is 3.04905. Thus, by adding a third member, the total weight of the structure is reduced. It should be noted, however, that the behavior of the trusses will not be the same.

Switsky [4] shows that the deflection of the two-bar truss will be less in this case than that of the three-bar truss. If the displacements of both trusses are required to be equal, given the same stress constraints, the statically determinant two-bar truss will be the lighter of the two.

## C. NON-ANALYTICAL CONSTRAINTS

Structural problems often do not allow the nice problem statement form of the three-bar truss problem. The major difficulty arises from not being able to express some of the constraints in the completely analytical form required for the Fiacco-McCormick technique. Many problems require the use of empirically determined alignment charts or graphs to calculate changes in structural behavior as a result of the variation of design variables. For such problems, some variation of Schmit's "method of alternate steps" may be well suited. An example of such a problem is the determination of the minimum weight design of an integrally stiffened panel loaded axially under compression. Rosenbaum [5] describes the problem and outlines a solution procedure based on the method of alternate steps.

## VII. RECOMMENDATIONS FOR FURTHER STUDY

Since only three design variables of the three-bar truss are allowed to vary, the example is actually a sub-optimization problem. This fact suggests that several possible extensions of the problem merit further consideration.

Allowing more design variables such as the length of truss members to vary makes the problem even more nonconvex but the Fiacco-McCormick procedure may still be applied. The assumption of the availability of a continuous spectrum of materials would permit the determination of optimal material properties of the structure.

The added restriction of a discrete choice of materials or material sizes available is a further realistic extension of the problem although a different solution procedure is required.

Finally, the development of a systematic procedure for finding alternate minima is essential if one is to accept any result with confidence.

COMPUTER PROGRAM

```
C MAIN PROGRAM
C
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 K11,K12,K13,K21,K22,K23,K14,K15,K25,K31,K42,K53
     $,K34,K35,K44,K45,K54,K55,K24
      DIMENSION X(13),XX(13,12),XEST(13,12),X2(13),SPR(13,13
     $),COL(169),DELP(13),XOLD(13),LWK(13),MWK(13),C(13),H(1
     $C,12)
      DIMENSION RR(12),P(12),F(12),SIGMA(12),PEST(12),IA(12)
      DIMENSION XY(4),RY(4),PY(4)
      EQUIVALENCE(SPR(1),COL(1))
      COMMON C1,C2,C3,K11,K12,K13,K21,K22,K23,K14,K24,K15,K2
     $5,K31,K42,K53,K34,K35,K44,K45,K54,K55
C DDTOL - TOLERANCE FOR DIRECTIONAL DERIVATIVE.
C EPSI - TOLERANCE FOR GRADIENT MAGNITUDE
      DATA DDTOL/.0000001/,EPSI/.001/,DELTA/1./
C STARTING X VALUES FOR SEARCH.
      DATA X/3*2.0,10*5.0/
C NV - NUMBER OF VARIABLES.
      NV=13
      RR(1)=1.
C IR - NUMBER OF R VALUES FOR WHICH P FUNCTION IS TO BE
C MINIMIZED.
      IR=9
C J - NUMBER OF R VALUES FOR WHICH P HAS BEEN MINIMIZED.
      J=0
    5 J=J+1
C RR(J) - JTH R VALUE.
      R=RR(J)
C
C BEGIN SECOND ORDER GRADIENT SEARCH FOR MINIMUM OF P FOR
C CURRENT R VALUE.
C
C DDOLD - PREVIOUS DIRECTIONAL DERIVATIVE.
      DDOLD=0.C
C GOLD - PREVIOUS GRADIENT MAGNITUDE.
      GOLD=1.C
C XOLD - X VALUES CORRESPONDING TO PREVIOUS GRADIENT
C MAGNITUDE.
      DO 11 I=1,NV
      XOLD(I)=C.C
   11 CONTINUE
C II - COUNT OF MOVES IN SEARCH FOR MINIMUM OF P FOR CURRENT
C R VALUE.
      II=0
C EVALUATE HESSIAN MATRIX AT CURRENT STARTING POINT.
C SPR - HESSIAN MATRIX.
   13 CALL MSOPAR(X,R,SPR,NV)
C EVALUATE GRADIENT VECTOR AT CURRENT STARTING POINT.
C DELP - GRADIENT OF P.
      CALL GRAD(X,R,DELP)
C INVERT HESSIAN MATRIX.
      CALL MINV(COL,NV,D,LWK,MWK)
C STOP IF HESSIAN MATRIX IS SINGULAR. PICK NEW STARTING
C POINT.
      IF(D.EQ.C.C) WRITE(6,17)
   17 FORMAT('*DET OF MATRIX OF 2ND ORDER PARTIALS = 0*')
      IF(D.EQ.C.C) STOP
C PREMULTIPLY INVERSE OF HESSIAN BY GRADIENT VECTOR.
      CALL MTXMUL(SPR,DELP,NV,C)
C GMAG - MAGNITUDE OF GRADIENT.
      GMAG=0.C
      DO 23 I=1,NV
      GMAG=GMAG+(DELP(I)*DELP(I))
   23 CONTINUE
      GMAG=DSQRT(GMAG)
C TEST MAGNITUDE OF GRADIENT. IF BELOW TOLERANCE, P IS
C MINIMIZED FOR CURRENT R.
      IF(GMAG.GT.EPSI) GO TO 28
```

```
C PSTAR - P MINIMUM FOR CURRENT R VALUE.
      PSTAR=PVALUE(X,R)
      GO TO 85
C IF GRADIENT MAGNITUDE IS SMALL (LESS THAN DELTA) AND
C GMAG INCREASES, CONVERGENCE IS UNCERTAIN. TERMINATE
C SEARCH WITH SMALLEST GRADIENT MAGNITUDE OBTAINED.
   28 IF(GMAG.LT.DELTA.AND.GMAG.GE.GOLD) GO TO 30
      GO TO 38
   30 WRITE(6,31) GOLD
   31 FORMAT(' *MAGNITUDE OF GRADIENT WILL NOT CONVERGE TO '
     $'WITHIN TOLERANCE. LAST GMAG OBTAINED WAS'F10.5'*'/)
      DO 35 I=1,NV
      X(I)=XOLD(I)
   35 CONTINUE
      PSTAR=PVALUE(X,R)
      GO TO 85
C SAVE PRESENT GRADIENT MAGNITUDE AND LOCATION FOR NEXT
C CONVERGENCE TEST.
   38 GOLD=GMAG
      DO 41 I=1,NV
      XOLD(I)=X(I)
   41 CONTINUE
C EVALUATE DIRECTIONAL DERIVATIVE AT STARTING POINT.
C DD - DIRECTIONAL DERIVATIVE.
      DD=DIRDER(NV,DELP,C)
C IF DIRECTIONAL DERIVATIVE IS WITHIN TOLERANCE BUT GRADIENT
C IS NOT, CONVERGENCE IS IMPOSSIBLE WITH PRESENT TOLERANCES.
      IF(DABS(DD).GT.DDTOL) GO TO 48
      WRITE(6,45) DD,R,II
   45 FORMAT(' *EPSI TOO SMALL OR DDTOL TOO LARGE. SEARCH W'
     $'ILL NOT CONVERGE. LAST DD OBTAINED ='F12.8'.R='F10.8
     $'. I = 'I4'*')
      STOP
C
C BEGIN GRADIENT SEARCH FOR MINIMUM IN CURRENT DIRECTION OF
C STEEPEST DESCENT.
C
C UNITIZE DIRECTIONAL DERIVATIVE. KEEP SIGN. EVALUATE P
C FUNCTION AT START POINT.
   48 TEST=DD/DABS(DD)
      PV1=PVALUE(X,R)
C INITIAL THETA VALUE IS POSITIVE.
      T1=0.0
      STEP=.5
   52 T=T1+STEP
C STEP IN DIRECTION OF STEEPEST DESCENT TO GET X2.
      CALL XVALUE(X2,X,NV,C,T)
C TEST X2 FOR FEASIBILITY WITH RESPECT TO INEQUALITY
C CONSTRAINTS.
      CALL CHECK(X2,NV,RESULT)
C IF RESULT IS ONE, X2 IS FEASIBLE. IF X2 IS NOT FEASIBLE,
C REDUCE THETA VALUE.
      IF(RESULT.EQ.1.0) GO TO 58
      STEP=STEP/2.
      GO TO 52
   58 T1=T
C EVALUATE P AFTER STEP.
      PV2=PVALUE(X2,R)
C EVALUATE DIRECTIONAL DERIVATIVE AT X2.
      CALL GRAD(X2,R,DELP)
      DD=DIRDER(NV,DELP,C)
C TEST FOR MINIMUM AT X2.
      IF(DABS(DD).LE.DDTOL) GO TO 82
C TEST SIGN OF DIRECTIONAL DERIVATIVE AT X2. IF OPPOSITE
C FROM THAT AT STARTING POINT, BRACKET HAS BEEN ESTABLISHED.
      TRY=DD/DABS(DD)
      IF(TRY+TEST) 65,68,65
C IF NOT OPPOSITE IN SIGN BUT P INCREASES, SEARCH FOR
C MINIMUM IN DIRECTION OF NEGATIVE THETA FROM START POINT.
   65 IF(PV2.LT.PV1) GO TO 52
      STEP=-2.
      GO TO 52
```

```
C INITIAL BRACKET IS ESTABLISHED. REDUCE STEP SIZE BY 1/2.
C REVERSE DIRECTION OF SEARCH WITH REDUCED STEP SIZE.
   68 TEST=TRY
      STEP=T
   70 STEP=-STEP/2.
   71 T=T+STEP
      CALL XVALUE(X2,X,NV,C,T)
      CALL GRAD(X2,R,DELP)
      DD=DIRDER(NV,DELP,C)
C TEST FOR MINIMUM AFTER EACH STEP.
      IF(DABS(DD).LE.DDTOL) GO TO 82
C TERMINATE SEARCH IF DIRECTIONAL DERIVATIVE REPEATS.
      IF(DD.EQ.DDOLD) GO TO 82
C SAVE CURRENT VALUE OF DD.
      DDOLD=DD
C TEST FOR ESTABLISHMENT OF REDUCED BRACKET.
      TRY=DD/DABS(DD)
      IF(TRY+TEST) 71,80,71
   80 TEST=TRY
      GO TO 70
C CALCULATE VALUE OF NEW STARTING POINT. BEGIN SEARCH IN
C NEW DIRECTION OF STEEPEST DESCENT.
   82 CALL XVALUE(X,X,NV,C,T)
      II=II+1
      GO TO 13
   85 IA(J)=II
      P(J)=PSTAR
C EVALUATE ORIGINAL OBJECTIVE FUNCTION.
      F(J)=X(1)*C1+X(2)*C2+X(3)*C3
      SIGMA(J)=P(J)-F(J)
C EVALUATE EQUALITY CONSTRAINTS.
      H(1,J)=X(1)*X(4)*K11+X(2)*X(5)*K12+X(3)*X(6)*K13-K14
      H(2,J)=X(1)*X(4)*K21+X(2)*X(5)*K22+X(3)*X(6)*K23-K24
      H(3,J)=K31*X(4)+K34*X(7)+K35*X(8)
      H(4,J)=K42*X(5)+K44*X(7)+K45*X(8)
      H(5,J)=K53*X(6)+K54*X(7)+K55*X(8)
      H(6,J)=X(1)*X(9)*K11+X(2)*X(10)*K12+X(3)*X(11)*K13-K15
      H(7,J)=X(1)*X(9)*K21+X(2)*X(10)*K22+X(3)*X(11)*K23-K25
      H(8,J)=K31*X(9)+K34*X(12)+K35*X(13)
      H(9,J)=K42*X(10)+K44*X(12)+K45*X(13)
      H(10,J)=K53*X(11)+K54*X(12)+K55*X(13)
      DO 92 K=1,NV
      XX(K,J)=X(K)
   92 CONTINUE
      IF(J.LT.4) GO TO 108
C
C BEGIN THIRD ORDER POLYNOMIAL CURVE FIT TO ESTIMATE P AND X
C VALUES FOR R=0. PY,RY,AND XX CONTAIN LAST FOUR VALUES OF
C P,R,AND X IN FORM REQUIRED FOR SUBROUTINE POLY.
C
      I1=0
      J3=J-3
      DO 100 K=J3,J
      I1=I1+1
      PY(I1)=P(K)
      RY(I1)=RR(K)
  100 CONTINUE
      CALL POLY(PY,RY,4,A0)
C PEST - ESTIMATE OF P AT R=0.
      PEST(J)=A0
      DO 105 I=1,NV
      I1=0
      J3=J-3
      DO 102 K=J3,J
      I1=I1+1
      XY(I1)=XX(I,K)
  102 CONTINUE
      CALL POLY(XY,RY,4,A0)
C XEST - ESTIMATE OF X AT R=0.
      XEST(I,J)=A0
  105 CONTINUE
      IF(J.EQ.IR) GO TO 122
```

```
  108 J1=J+1
C REDUCE R BY ARBITRARY CONSTANT.
      RR(J1)=RR(J)/10.
      GO TO 5
  122 WRITE(6,123)
  123 FORMAT('1'44X'MOVES'/43X'REQUIRED'/46X'TO'/23X'LINE'8X
     $'R'7X'MINIMIZE'4X'P'8X'F'7X'P-F'8X'H1'8X'H2'8X'H3')
      DO 133 J=1,IR
      WRITE(6,128)J,RR(J),IA(J),P(J),F(J),SIGMA(J),(H(I,J),I
     $=1,3)
  128 FORMAT('0'22X,I3,3X,F12.9,3X,I3,3X,F8.5,F9.5,F10.5,
     $F10.5,F10.5,F10.5)
      IF(J.LT.4) GO TO 133
      WRITE(6,132) PEST(J)
  132 FORMAT(' '29X'3RD ORDER ESTIMATE  'F8.5)
  133 CONTINUE
      WRITE(6,135)
  135 FORMAT('1'23X'LINE'7X'R'10X'H4'8X'H5'8X'H6'8X'H7'8X'H8
     $8X'H9'8X'H10')
      WRITE(6,138)(J,RR(J),(H(I,J),I=4,10),J=1,IR)
  138 FORMAT('0'23X,I3,F14.9,7F10.6)
      WRITE(6,140)
  140 FORMAT('1'23X'LINE'7X'R'11X'X1'8X'X2'8X'X3'8X'X4'8X'X5
     $8X'X6'8X'X7')
      DO 148 J=1,IR
      WRITE(6,144)J,RR(J),(XX(I,J),I=1,7)
  144 FORMAT('0'23X,I3,F14.9,7F10.5)
      IF(J.LT.4) GO TO 148
      WRITE(6,147) (XEST(I,J),I=1,7)
  147 FORMAT(' ',28X'3RD ORDER EST'F9.5,7F10.5)
  148 CONTINUE
      WRITE(6,150)
  150 FORMAT('1'28X'LINE'7X'R'11X'X8'8X'X9'8X'X10'7X'X11'7X
     $'X12'7X'X13')
      DO 158 J=1,IR
      WRITE(6,154)J,RR(J),(XX(I,J),I=8,13)
  154 FORMAT('0'28X,I3,F14.9,6F10.5)
      IF(J.LT.4) GO TO 158
      WRITE(6,157)(XEST(I,J),I=8,13)
  157 FORMAT(' ',33X'3RD ORDER EST'F9.5,6F10.5)
  158 CONTINUE
      STOP
      END
C
C
C SUBPROGRAMS


      SUBROUTINE POLY(X,RR,K,AO)
C THIS SUBROUTINE SOLVES THE LINEAR EQUATIONS FOR THE
C POLYNOMIAL ESTIMATE OF THE VARIABLE CONTAINED IN
C ARGUMENT X.  AO IS THE ESTIMATED VALUE OF THE VARIABLE
C AT R=0.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(4),RR(4),A(4,4),L(16),M(16),C(16)
      EQUIVALENCE(A(1),C(1))
      DO 1 J=1,K
      DO 1 I=1,K
      A(I,J)=(DSQRT(RR(I)))**(J-1)
    1 CONTINUE
      CALL MINV(C,K,D,L,M)
      AO=0.0
      DO 2 J1=1,K
      AO=AO+(A(1,J1))*X(J1)
    2 CONTINUE
      RETURN
      END
C
C
      DOUBLE PRECISION FUNCTION DIRDER(NV,DELP,C)
C THIS SUBPROGRAM CALCULATES THE VALUE OF THE DIRECTIONAL
C DERIVATIVE.
```

```
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 DELP(NV),C(NV),MAGSQ
      MAGSQ=0.0
      DO 1 I=1,NV
      MAGSQ=MAGSQ+(C(I)*C(I))
    1 CONTINUE
      IF (MAGSQ.EQ.0.0) GO TO 3
      UV=1/DSQRT(MAGSQ)
      DIRDER=0.0
      DO 2 I=1,NV
      DIRDER=DIRDER+(DELP(I)*C(I)*UV)
    2 CONTINUE
      RETURN
    3 DIRDER =0.0
      RETURN
      END


      SUBROUTINE MTXMUL(SPR,DELP,NV,C)
C THIS SUBROUTINE PREMULTIPLIES THE VECTOR DELP BY THE
C MATRIX SPR.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION SPR(NV,NV),DELP(NV),C(NV)
      DO 1 I=1,NV
      C(I)=0.0
      DO 1 J=1,NV
      C(I)=C(I)+(SPR(I,J)*DELP(J))
    1 CONTINUE
      RETURN
      END


      SUBROUTINE XVALUE (X2,X1,J,C,T)
C THIS SUBROUTINE CALCULATES THE POINT X2 A DISTANCE
C PROPORTIONAL TO T FROM X1 IN THE DIRECTION OF
C STEEPEST DESCENT.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X2(J),X1(J),C(J,1)
      DO 1 I = 1,J
      X2(I) = X1(I)-(T*C(I,1))
    1 CONTINUE
      RETURN
      END


      SUBROUTINE CHECK(X,NOVAR,RESULT)
C THIS SUBROUTINE CONTAINS THE INEQUALITY CONSTRAINTS.
C IF THE POINT X SATISFIES THESE CONSTRAINTS, RESULT=+1.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(13),L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,K11,K12
     $,K13,K21,K22,K23,K14,K15,K25,K31,K42,K53,K34,K35,K44,
     $K45,K54,K55,K24
      COMMON C1,C2,C3,K11,K12,K13,K21,K22,K23,K14,K24,K15,
     $K25,K31,K42,K53,K34,K35,K44,K45,K54,K55,U4,U5,U6,U7,U8
     $,L4,L5,L6,L7,L8
      EQUIVALENCE (L4,L9),(L5,L10),(L6,L11),(L7,L12),(L8,L13
     $),(U4,U9),(U5,U10),(U6,U11),(U7,U12),(U8,U13)
      IF(X(1).LE.0.0) GO TO 1
      IF(X(2).LE.0.0) GO TO 1
      IF(X(3).LE.0.0) GO TO 1
      IF(X(4).LE.L4.OR.X(4).GE.U4) GO TO 1
      IF(X(5).LE.L5.OR.X(5).GE.U5) GO TO 1
      IF(X(6).LE.L6.OR.X(6).GE.U6) GO TO 1
      IF(X(7).LE.L7.OR.X(7).GE.U7) GO TO 1
      IF(X(8).LE.L8.OR.X(8).GE.U8) GO TO 1
      IF(X(9).LE.L9.OR.X(9).GE.U9) GO TO 1
      IF(X(10).LE.L10.OR.X(10).GE.U10) GO TO 1
      IF(X(11).LE.L11.OR.X(11).GE.U11) GO TO 1
      IF(X(12).LE.L12.OR.X(12).GE.U12) GO TO 1
      IF(X(13).LE.L13.OR.X(13).GE.U13) GO TO 1
      RESULT=1.0
      RETURN
```

```
    1 RESULT=-1.0
      RETURN
      END
      DOUBLE PRECISION FUNCTION PVALUE(X,R)
C THIS SUBPROGRAM CONTAINS THE ANALYTICAL FORM OF THE
C P-FUNCTION.  THE VALUE OF P AT X IS CALCULATED.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 X(13),L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,K11,K12
     $,K13,K21,K22,K23,K14,K15,K25,K31,K42,K53,K34,K35,K44,
     $K45,K54,K55,K24
      COMMON C1,C2,C3,K11,K12,K13,K21,K22,K23,K14,K24,K15,
     $K25,K31,K42,K53,K34,K35,K44,K45,K54,K55,U4,U5,U6,U7,U8
     $,L4,L5,L6,L7,L8
      EQUIVALENCE (L4,L9),(L5,L10),(L6,L11),(L7,L12),(L8,L13
     $),(U4,U9),(U5,U10),(U6,U11),(U7,U12),(U8,U13)
      H1=X(1)*X(4)*K11+X(2)*X(5)*K12+X(3)*X(6)*K13-K14
      H2=X(1)*X(4)*K21+X(2)*X(5)*K22+X(3)*X(6)*K23-K24
      H3=K31*X(4)+K34*X(7)+K35*X(8)
      H4=K42*X(5)+K44*X(7)+K45*X(8)
      H5=K53*X(6)+K54*X(7)+K55*X(8)
      H6=X(1)*X(9)*K11+X(2)*X(10)*K12+X(3)*X(11)*K13-K15
      H7=X(1)*X(9)*K21+X(2)*X(10)*K22+X(3)*X(11)*K23-K25
      H8=K31*X(9)+K34*X(12)+K35*X(13)
      H9=K42*X(10)+K44*X(12)+K45*X(13)
      H10=K53*X(11)+K54*X(12)+K55*X(13)
      G1=1./X(1)
      G2=1./X(2)
      G3=1./X(3)
      G4=1./(X(4)-L4)
      G5=1./(X(4)-U4)
      G6=1./(X(5)-L5)
      G7=1./(X(5)-U5)
      G8=1./(X(6)-L6)
      G9=1./(X(6)-U6)
      G10=1./(X(7)-L7)
      G11=1./(X(7)-U7)
      G12=1./(X(8)-L8)
      G13=1./(X(8)-U8)
      G14=1./(X(9)-L9)
      G15=1./(X(9)-U9)
      G16=1./(X(10)-L10)
      G17=1./(X(10)-U10)
      G18=1./(X(11)-L11)
      G19=1./(X(11)-U11)
      G20=1./(X(12)-L12)
      G21=1./(X(12)-U12)
      G22=1./(X(13)-L13)
      G23=1./(X(13)-U13)
      Y1=1./DSQRT(R)
      EQ=H1**2+H2**2+H3**2+H4**2+H5**2+H6**2+H7**2+H8**2+
     $H9**2+H10**2
      UN=G1+G2+G3+G4-G5+G6-G7+G8-G9+G10-G11+G12-G13+G14-G15+
     $G16-G17+G18-G19+G20-G21+G22-G23
      PVALUE=C1*X(1)+C2*X(2)+C3*X(3)+Y1*EQ+R*UN
      RETURN
      END


      SUBROUTINE GRAD(X,R,D)
C THIS SUBPROGRAM CONTAINS THE ANALYTICAL FORM OF THE
C GRADIENT VECTOR.  THE VECTOR IS EVALUATED AT X.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 D(13)
      REAL*8 X(13),L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,K11,K12
     $,K13,K21,K22,K23,K14,K15,K25,K31,K42,K53,K34,K35,K44,
     $K45,K54,K55,K24
      COMMON C1,C2,C3,K11,K12,K13,K21,K22,K23,K14,K24,K15,
     $K25,K31,K42,K53,K34,K35,K44,K45,K54,K55,U4,U5,U6,U7,U8
     $,L4,L5,L6,L7,L8
      EQUIVALENCE (L4,L9),(L5,L10),(L6,L11),(L7,L12),(L8,L13
     $),(U4,U9),(U5,U10),(U6,U11),(U7,U12),(U8,U13)
      DIMENSION RO(3),E(3),BA(3),AA(2),P(2)
```

```
      H1=X(1)*X(4)*K11+X(2)*X(5)*K12+X(3)*X(6)*K13-K14
      H2=X(1)*X(4)*K21+X(2)*X(5)*K22+X(3)*X(6)*K23-K24
      H3=K31*X(4)+K34*X(7)+K35*X(8)
      H4=K42*X(5)+K44*X(7)+K45*X(8)
      H5=K53*X(6)+K54*X(7)+K55*X(8)
      H6=X(1)*X(9)*K11+X(2)*X(10)*K12+X(3)*X(11)*K13-K15
      H7=X(1)*X(9)*K21+X(2)*X(10)*K22+X(3)*X(11)*K23-K25
      H8=K31*X(9)+K34*X(12)+K35*X(13)
      H9=K42*X(10)+K44*X(12)+K45*X(13)
      H10=K53*X(11)+K54*X(12)+K55*X(13)
      G1=1./X(1)
      G2=1./X(2)
      G3=1./X(3)
      G4=1./(X(4)-L4)
      G5=1./(X(4)-U4)
      G6=1./(X(5)-L5)
      G7=1./(X(5)-U5)
      G8=1./(X(6)-L6)
      G9=1./(X(6)-U6)
      G10=1./(X(7)-L7)
      G11=1./(X(7)-U7)
      G12=1./(X(8)-L8)
      G13=1./(X(8)-U8)
      G14=1./(X(9)-L9)
      G15=1./(X(9)-U9)
      G16=1./(X(10)-L10)
      G17=1./(X(10)-U10)
      G18=1./(X(11)-L11)
      G19=1./(X(11)-U11)
      G20=1./(X(12)-L12)
      G21=1./(X(12)-U12)
      G22=1./(X(13)-L13)
      G23=1./(X(13)-U13)
      Y2=2./DSQRT(R)
      D(1)=C1+Y2*(X(4)*(K11*H1+K21*H2)+X(9)*(K11*H6+K21*H7))
     $-R*(G1**2)
      D(2)=C2+Y2*(X(5)*(K12*H1+K22*H2)+X(10)*(K12*H6+K22*H7)
     $)-R*(G2**2)
      D(3)=C3+Y2*(X(6)*(K13*H1+K23*H2)+X(11)*(K13*H6+K23*H7)
     $)-R*(G3**2)
      D(4)=Y2*(X(1)*(K11*H1+K21*H2)+K31*H3)-R*(G4**2-G5**2)
      D(5)=Y2*(X(2)*(K12*H1+K22*H2)+K42*H4)-R*(G6**2-G7**2)
      D(6)=Y2*(X(3)*(K13*H1+K23*H2)+K53*H5)-R*(G8**2-G9**2)
      D(7)=Y2*(K34*H3+K44*H4+K54*H5)-R*(G10**2-G11**2)
      D(8)=Y2*(K35*H3+K45*H4+K55*H5)-R*(G12**2-G13**2)
      D(9)=Y2*(X(1)*(K11*H6+K21*H7)+K31*H8)-R*(G14**2-G15**2
     $)
      D(10)=Y2*(X(2)*(K12*H6+K22*H7)+K42*H9)-R*(G16**2-
     $G17**2)
      D(11)=Y2*(X(3)*(K13*H6+K23*H7)+K53*H10)-R*(G18**2-
     $G19**2)
      D(12)=Y2*(K34*H8+K44*H9+K54*H10)-R*(G20**2-G21**2)
      D(13)=Y2*(K35*H8+K45*H9+K55*H10)-R*(G22**2-G23**2)
      RETURN
      END


      SUBROUTINE MSOPAR(X,R,S,NV)
C THIS SUBROUTINE CONTAINS THE ANALYTICAL FORM OF THE
C HESSIAN MATRIX.  THE MATRIX IS EVALUATED AT X.
C PROBLEM CONSTANTS ARE READ INTO COMMON IN THIS ROUTINE.
      IMPLICIT REAL*8(A-H,O-Z)
      REAL*8 S(13,13)
      REAL*8 N
      REAL*8 X(13),L4,L5,L6,L7,L8,L9,L10,L11,L12,L13,K11,K12
     $,K13,K21,K22,K23,K14,K15,K25,K31,K42,K53,K34,K35,K44,
     $K45,K54,K55,K24
      COMMON C1,C2,C3,K11,K12,K13,K21,K22,K23,K14,K24,K15,
     $K25,K31,K42,K53,K34,K35,K44,K45,K54,K55,U4,U5,U6,U7,U8
     $,L4,L5,L6,L7,L8
      EQUIVALENCE (L4,L9),(L5,L10),(L6,L11),(L7,L12),(L8,L13
     $),(U4,U9),(U5,U10),(U6,U11),(U7,U12),(U8,U13)
```

```
      DIMENSION RO(3),E(3),BA(3),AA(2),P(2)
      DATA RO(1),RO(2),RO(3),E(1),E(2),E(3),N/7*1.0/
      DATA L/0/
      L=L+1
      IF(L.GT.1) GO TO 2
      U4=20.0
      U5=20.0
      U6=20.0
      U7=200.
      U8=200.
      L4=-15.
      L5=-15.
      L6=-15.
      L7=-200.
      L8=-200.
      PI=3.1415926536
      BA(1)=.75*PI
      BA(2)=.50*PI
      BA(3)=.25*PI
      AA(1)=PI/3.
      AA(2)=PI
      P(1)=30.
      P(2)=20.
      C1=N*RO(1)/DSIN(BA(1))
      C2=N*RO(2)/DSIN(BA(2))
      C3=N*RO(3)/DSIN(BA(3))
      K11=DCOS(BA(1))
      K12=DCOS(BA(2))
      K13=DCOS(BA(3))
      K21=DSIN(BA(1))
      K22=DSIN(BA(2))
      K23=DSIN(BA(3))
      K14=-P(1)*DCOS(AA(1))
      K24=P(1)*DSIN(AA(1))
      K15=-P(2)*DCOS(AA(2))
      K25=P(2)*DSIN(AA(2))
      K31=N/(E(1)*DSIN(BA(1)))
      K42=N/(E(2)*DSIN(BA(2)))
      K53=N/(E(3)*DSIN(BA(3)))
      K34=K11
      K35=K21
      K44=K12
      K45=K22
      K54=K13
      K55=K23
    2 Y2=2./DSQRT(R)
      H1=X(1)*X(4)*K11+X(2)*X(5)*K12+X(3)*X(6)*K13-K14
      H2=X(1)*X(4)*K21+X(2)*X(5)*K22+X(3)*X(6)*K23-K24
      H3=K31*X(4)+K34*X(7)+K35*X(8)
      H4=K42*X(5)+K44*X(7)+K45*X(8)
      H5=K53*X(6)+K54*X(7)+K55*X(8)
      H6=X(1)*X(9)*K11+X(2)*X(10)*K12+X(3)*X(11)*K13-K15
      H7=X(1)*X(9)*K21+X(2)*X(10)*K22+X(3)*X(11)*K23-K25
      H8=K31*X(9)+K34*X(12)+K35*X(13)
      H9=K42*X(10)+K44*X(12)+K45*X(13)
      H10=K53*X(11)+K54*X(12)+K55*X(13)
      G1=1./X(1)
      G2=1./X(2)
      G3=1./X(3)
      G4=1./(X(4)-L4)
      G5=1./(X(4)-U4)
      G6=1./(X(5)-L5)
      G7=1./(X(5)-U5)
      G8=1./(X(6)-L6)
      G9=1./(X(6)-U6)
      G10=1./(X(7)-L7)
      G11=1./(X(7)-U7)
      G12=1./(X(8)-L8)
      G13=1./(X(8)-U8)
      G14=1./(X(9)-L9)
      G15=1./(X(9)-U9)
      G16=1./(X(10)-L10)
```

```
      G17=1./(X(10)-U10)
      G18=1./(X(11)-L11)
      G19=1./(X(11)-U11)
      G20=1./(X(12)-L12)
      G21=1./(X(12)-U12)
      G22=1./(X(13)-L13)
      G23=1./(X(13)-U13)
      S(1,1)=Y2*(X(4)**2+X(9)**2)*(K11**2+K21**2)+2.*R*G1**3
      S(1,2)=Y2*(X(4)*X(5)+X(9)*X(10))*(K11*K12+K21*K22)
      S(1,3)=Y2*(X(4)*X(6)+X(9)*X(11))*(K11*K13+K21*K23)
      S(1,4)=Y2*(K11*H1+K21*H2+X(1)*X(4)*(K11**2+K21**2))
      S(1,5)=Y2*X(2)*X(4)*(K11*K12+K21*K22)
      S(1,6)=Y2*X(3)*X(4)*(K11*K13+K21*K23)
      S(1,7)=0.0
      S(1,8)=0.0
      S(1,9)=Y2*(K11*H6+K21*H7+X(1)*X(9)*(K11**2+K21**2))
      S(1,10)=Y2*(X(2)*X(9)*(K11*K12+K21*K22))
      S(1,11)=Y2*(X(3)*X(9)*(K11*K13+K21*K23))
      S(1,12)=0.0
      S(1,13)=0.0
      S(2,2)=Y2*(X(5)**2+X(10)**2)*(K12**2+K22**2)+2.*R*
     $G2**3
      S(2,3)=Y2*(X(5)*X(6)+X(10)*X(11))*(K12*K13+K22*K23)
      S(2,4)=Y2*X(1)*X(5)*(K11*K12+K21*K22)
      S(2,5)=Y2*(K12*H1+K22*H2+X(2)*X(5)*(K12**2+K22**2))
      S(2,6)=Y2*X(3)*X(5)*(K12*K13+K22*K23)
      S(2,7)=0.0
      S(2,8)=0.0
      S(2,9)=Y2*X(1)*X(10)*(K11*K12+K21*K22)
      S(2,10)=Y2*(K12*H6+K22*H7+X(2)*X(10)*(K12**2+K22**2))
      S(2,11)=Y2*X(3)*X(10)*(K12*K13+K22*K23)
      S(2,12)=0.0
      S(2,13)=0.0
      S(3,3)=Y2*(X(6)**2+X(11)**2)*(K13**2+K23**2)+2.*R*
     $G3**3
      S(3,4)=Y2*X(1)*X(6)*(K11*K13+K21*K23)
      S(3,5)=Y2*X(2)*X(6)*(K12*K13+K22*K23)
      S(3,6)=Y2*(K13*H1+K23*H2+X(3)*X(6)*(K13**2+K23**2))
      S(3,7)=0.0
      S(3,8)=0.0
      S(3,9)=Y2*X(1)*X(11)*(K11*K13+K21*K23)
      S(3,10)=Y2*X(2)*X(11)*(K12*K13+K22*K23)
      S(3,11)=Y2*(K13*H6+K23*H7+X(3)*X(11)*(K13**2+K23**2))
      S(3,12)=0.0
      S(3,13)=0.0
      S(4,4)=Y2*((X(1)**2)*(K11**2+K21**2)+K31**2)+2.*R*
     $(G4**3-G5**3)
      S(4,5)=Y2*X(1)*X(2)*(K11*K12+K21*K22)
      S(4,6)=Y2*X(1)*X(3)*(K11*K13+K21*K23)
      S(4,7)=Y2*K31*K34
      S(4,8)=Y2*K31*K35
      S(4,9)=0.0
      S(4,10)=0.0
      S(4,11)=0.0
      S(4,12)=0.0
      S(4,13)=0.0
      S(5,5)=Y2*((X(2)**2)*(K12**2+K22**2)+K42**2)+2.*R*
     $(G6**3-G7**3)
      S(5,6)=Y2*X(2)*X(3)*(K12*K13+K22*K23)
      S(5,7)=Y2*K42*K44
      S(5,8)=Y2*K42*K45
      S(5,9)=0.0
      S(5,10)=0.0
      S(5,11)=0.0
      S(5,12)=0.0
      S(5,13)=0.0
      S(6,6)=Y2*((X(3)**2)*(K13**2+K23**2)+K53**2)+2.*R*
     $(G8**3-G9**3)
      S(6,7)=Y2*K53*K54
      S(6,8)=Y2*K53*K55
      S(6,9)=0.0
      S(6,10)=0.0
```

```
      S(6,11)=0.0
      S(6,12)=0.0
      S(6,13)=0.0
      S(7,7)=Y2*(K34**2+K44**2+K54**2)+2.*R*(G10**3-G11**3)
      S(7,8)=Y2*(K34*K35+K44*K45+K54*K55)
      S(7,9)=0.0
      S(7,10)=0.0
      S(7,11)=0.0
      S(7,12)=0.0
      S(7,13)=0.0
      S(8,8)=Y2*(K35**2+K45**2+K55**2)+2.*R*(G12**3-G13**3)
      S(8,9)=0.0
      S(8,10)=0.0
      S(8,11)=0.0
      S(8,12)=0.0
      S(8,13)=0.0
      S(9,9)=Y2*((X(1)**2)*(K11**2+K21**2)+K31**2)+2.*R*
     $(G14**3-G15**3)
      S(9,10)=Y2*X(1)*X(2)*(K11*K12+K21*K22)
      S(9,11)=Y2*X(1)*X(3)*(K11*K13+K21*K23)
      S(9,12)=Y2*K31*K34
      S(9,13)=Y2*K31*K35
      S(10,10)=Y2*((X(2)**2)*(K12**2+K22**2)+K42**2)+2.*R*
     $(G16**3-G17**3)
      S(10,11)=Y2*X(2)*X(3)*(K12*K13+K22*K23)
      S(10,12)=Y2*K42*K44
      S(10,13)=Y2*K42*K45
      S(11,11)=Y2*((X(3)**2)*(K13**2+K23**2)+K53**2)+2.*R*
     $(G18**3-G19**3)
      S(11,12)=Y2*K53*K54
      S(11,13)=Y2*K53*K55
      S(12,12)=Y2*(K34**2+K44**2+K54**2)+2.*R*(G20**3-G21**3
     $)
      S(12,13)=Y2*(K34*K35+K44*K45+K54*K55)
      S(13,13)=Y2*(K35**2+K45**2+K55**2)+2.*R*(G22**3-G23**3
     $)
      DO 1 I=1,NV
      DO 1 J=1,NV
      S(J,I)=S(I,J)
    1 CONTINUE
      RETURN
      END


      SUBROUTINE MINV(A,N,D,L,M)
      DIMENSION A(1),L(1),M(1)
      DOUBLE PRECISION A,D,BIGA,HOLD
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND
C               REPLACED BY RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARD GAUSS-JORDAN METHOD IS USED. THE
C           DETERMINANT IS ALSO CALCULATED. A DETERMINANT OF
C           ZERO INDICATES THAT THE MATRIX IS SINGULAR.
```

```
C
C
C        SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF( DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT
C        ELEMENT IS CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
```

```
C           DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C           PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C           REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C           FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END
```

## BIBLIOGRAPHY

1. Schmit, L. A., Structural Design by Systematic Synthesis, Proceedings of the Second Conference on Electronic Computation, ASCE, p.105-132, September, 1960.

2. Research Analysis Corporation Technical Paper RAC-TP-155, The Sequential Unconstrained Minimization Technique for Convex Programming with Equality Constraints, by A. V. Fiacco and G. P. McCormick, April, 1965.

3. Research Analysis Corporation Technical Paper RAC-TP-96, Programming under Nonlinear Constraints by Unconstrained Minimization: A Primal-Dual Method, by A. V. Fiacco and G. P. McCormick, September, 1963.

4. Switzky, H., "Designing for Minimum Flexibility of Weight," Journal of Spacecraft, Vol. 5, p.1473-1475, December, 1968.

5. Grumman Aircraft Engin. Co., Report No. ADR 02-21-65.1, Structural Synthesis of an Integrally Stiffened Panel Subjected to Compressive Load, by J. Rosenbaum, A. Kelley, and H. Pardo, March, 1965.

6. Fiacco, A. V., and McCormick, G. P., "Extensions of SUMT for Nonlinear Programming: Equality Constraints and Extra-polation," Vol. 12, No. 11, p.816-828, July, 1966.

7. Hadley, G., Nonlinear and Dynamic Programming, Addison-Wesley, 1964.

## INITIAL DISTRIBUTION LIST

| | | No. Copies |
|---|---|---|
| 1. | Defense Documentation Center<br>Cameron Station<br>Alexandria, Virginia 22314 | 20 |
| 2. | Library, Code 0212<br>Naval Postgraduate School<br>Monterey, California 93940 | 2 |
| 3. | Civil Schools Branch<br>Office of Personnel Operations<br>Department of the Army<br>Washington, D. C. 20315 | 1 |
| 4. | Professor Alan W. McMasters (Code 55Mg)<br>Department of Operations Analysis<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |
| 5. | Major William A. Crockett, USA<br>CDCMA<br>Aberdeen Proving Grounds, Md. 21005 | 1 |
| 6. | Professor Ulrich Haupt (Code 57Hp)<br>Department of Aeronautics<br>Naval Postgraduate School<br>Monterey, California 93940 | 1 |

Security Classification

# DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

| | |
|---|---|
| 1. ORIGINATING ACTIVITY (Corporate author)<br>Naval Postgraduate School<br>Monterey, California 93940 | 2a. REPORT SECURITY CLASSIFICATION<br>Unclassified<br>2b. GROUP |

3. REPORT TITLE

Optimal Design of Trussed Structures

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; October 1969

5. AUTHOR(S) (First name, middle initial, last name)

William Arthur Crockett

| 6. REPORT DATE | 7a. TOTAL NO. OF PAGES | 7b. NO. OF REFS |
|---|---|---|
| October 1969 | 49 | 7 |

| | |
|---|---|
| 8a. CONTRACT OR GRANT NO. | 9a. ORIGINATOR'S REPORT NUMBER(S) |
| b. PROJECT NO. | |
| c. | 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report) |
| d. | |

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

| 11. SUPPLEMENTARY NOTES | 12. SPONSORING MILITARY ACTIVITY |
|---|---|
| | Naval Postgraduate School<br>Monterey, California 93940 |

13. ABSTRACT

Structural design problems can be considered to be optimization problems because a design is sought which is optimal by some criterion subject to limitations on size, behavior, or other aspects of the structure. Under certain conditions, such problems may be solved by conventional mathematical programming techniques. The minimum weight design of a statically indeterminate three-bar truss is used to illustrate the application of the "sequential unconstrained minimization technique" of Fiacco and McCormick to the optimal design of trussed structures. A suggested computational procedure and computer program are included.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811



Security Classification

A-31408

Security Classification

| 14 KEY WORDS | LINK A | | LINK B | | LINK C | |
|---|---|---|---|---|---|---|
| | ROLE | WT | ROLE | WT | ROLE | WT |
| Truss | | | | | | |
| Structural Design | | | | | | |
| Nonlinear Programming | | | | | | |

DD FORM 1 NOV 65 1473 (BACK)

S/N 0101-807-6821



Security Classification

A-31409